


```
DDDDDDDD  RRRRRRR  7777777  8888888  0000000
DDDDDDDD  RRRRRRR  7777777  8888888  0000000
DD      DD  RR      RR      77      88      88      00      00
DD      DD  RR      RR      77      88      88      00      00
DD      DD  RR      RR      77      88      88      00      00
DD      DD  RR      RR      77      88      88      00      00
DD      DD  RRRRRRR  77      8888888  00      00      00
DD      DD  RRRRRRR  77      8888888  00      00      00
DD      DD  RR      RR      77      88      88      0000      00
DD      DD  RR      RR      77      88      88      0000      00
DD      DD  RR      RR      77      88      88      0000      00
DD      DD  RR      RR      77      88      88      0000      00
DDDDDDDD  RR      RR  77      8888888  0000000  0000000
DDDDDDDD  RR      RR  77      8888888  0000000  0000000
                                     ....
                                     ....
                                     ....
                                     ....
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```


F 10
16-Sep-1984 00:01:25
5-Sep-1984 13:53:26

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]DR780.FOR;1

Page 1

```
0001      SUBROUTINE DR780 (LUN)
0002      C
0003      C Version:      'V04-000'
0004      C
0005      C*****
0006      C*
0007      C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0008      C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0009      C*  ALL RIGHTS RESERVED.
0010      C*
0011      C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0012      C*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0013      C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0014      C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0015      C*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0016      C*  TRANSFERRED.
0017      C*
0018      C*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0019      C*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0020      C*  CORPORATION.
0021      C*
0022      C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0023      C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0024      C*
0025      C*
0026      C*****
0027      C
0028      C
0029      C
0030      C      AUTHOR  BRIAN PORTER          CREATION DATE  8-MAY-1979
0031      C
0032      C
0033      C++
0034      C      Functional description:
0035      C
0036      C      This module display entries for the DR780.
0037      C
0038      C      Modified by:
0039      C
0040      C      V03-004 SAR0220      Sharon A. Reynolds,      28-Mar-1984
0041      C      Changed the call to UCB$$_OWNUIC to ORB$$_OWNER.
0042      C
0043      C      V03-003 SAR0071      Sharon A. Reynolds,      20-Jun-1983
0044      C      Changed the carriage control in the 'format' statements
0045      C      for use with ERF.
0046      C
0047      C      V03-002 SAR0038      Sharon A. Reynolds,      8-Jun-1983
0048      C      Removed brief/cryptic support.
0049      C
0050      C      v03-001 BP0003      Brian Porter,      20-AUG-1982
0051      C      Minor edit.
0052      C**
0053      C--
0054      C
0055      C      Include 'SRC$:DR32COM.FOR /NOLIST'
0106      C      INCLUDE 'SRC$:MSGHDR.FOR /NOLIST'
0165      C      INCLUDE 'SRC$:DEVERR.FOR /NOLIST'
```

```

0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322

```

BYTE	LUN
INTEGER*4	FIELD
INTEGER*4	FIELD1
INTEGER*4	COMPRESSC
REAL	FLOAT
INTEGER*4	COMPRESS4
INTEGER*4	COMPRESSF
INTEGER*4	DR_CR
INTEGER*4	DR_UTL
INTEGER*4	DR_WCS(1:3)
INTEGER*4	DR_BA
INTEGER*4	DR_BC(1:2)
INTEGER*4	DR_CA(1:2)
INTEGER*4	DR_WD(1:5)
INTEGER*4	DR_SL
INTEGER*4	DR_SBR
INTEGER*4	DR_GBR
INTEGER*4	DR_CB(1:3)
INTEGER*4	DR_BB(1:3)
INTEGER*4	DR_GPR(0:15)
EQUIVALENCE	(DR_CR,EMBSL_DV_REGSAV(0))
EQUIVALENCE	(DR_UTL,EMBSL_DV_REGSAV(1))
EQUIVALENCE	(DR_WCS,EMBSL_DV_REGSAV(2))
EQUIVALENCE	(DR_BA,EMBSL_DV_REGSAV(5))
EQUIVALENCE	(DR_BC,EMBSL_DV_REGSAV(6))
EQUIVALENCE	(DR_CA,EMBSL_DV_REGSAV(9))
EQUIVALENCE	(DR_WD,EMBSL_DV_REGSAV(11))
EQUIVALENCE	(DR_SL,EMBSL_DV_REGSAV(16))
EQUIVALENCE	(DR_SBR,EMBSL_DV_REGSAV(17))
EQUIVALENCE	(DR_GBR,EMBSL_DV_REGSAV(18))
EQUIVALENCE	(DR_CB,EMBSL_DV_REGSAV(19))
EQUIVALENCE	(DR_BB,EMBSL_DV_REGSAV(22))
EQUIVALENCE	(DR_GPR,EMBSL_DV_REGSAV(25))
PARAMETER	TIMEOUT = 96
PARAMETER	DR_780 = 48
PARAMETER	SBI_FUNC1 = 1
PARAMETER	SBI_FUNC2 = 2
PARAMETER	SBI_FUNC7 = 7
PARAMETER	SBI_FUNC11 = 11
CHARACTER*10	V1DR_UTL(11:11)
DATA	V1DR_UTL(11) = '/'WCS VALID*'/
CHARACTER*31	V2DR_UTL(25:31)
DATA	V2DR_UTL(25) = '/'FORCE DDI CONTROL PARITY ERROR*'/
DATA	V2DR_UTL(26) = '/'FORCE DDI DATA PARITY ERROR*'/
DATA	V2DR_UTL(27) = '/'ENABLE DDI DATA PE ABORT*'/


```
0323      DATA      V2DR_UTL(28)      /*WCS PARITY ERROR*/
0324      DATA      V2DR_UTL(29)      /*DDI CONTROL PARITY ERROR*/
0325      DATA      V2DR_UTL(30)      /*DDI DATA PARITY ERROR*/
0326      DATA      V2DR_UTL(31)      /*PARITY ERROR*/
0327
0328
0329      C      DATA      V1DR_CFLAG(0) /*TRANSMIT INHIBIT SELECT*/
0330      C      DATA      V1DR_CFLAG(1) /*TRANSMIT MASTER STROBE*/
0331      C      DATA      V1DR_CFLAG(2) /*TRANSMIT SLAVE STROBE*/
0332      C      DATA      V1DR_CFLAG(3) /*TRANSMIT TO SLAVE*/
0333      C      DATA      V1DR_CFLAG(4) /*TRANSMIT RSEL*/
0334      C      DATA      V1DR_CFLAG(5) /*SWITCH ON DATA DRIVERS*/
0335
0336      C      DATA      V2DR_CFLAG(9) /*CSTAT*/
0337      C      DATA      V2DR_CFLAG(10) /*CIRDXP*/
0338      C      DATA      V2DR_CFLAG(11) /*CMDXP*/
0339      C      DATA      V2DR_CCFLAG(12) /*CIWDXP*/
0340
0341
0342      CHARACTER*3      BYTE_COUNT(1:2)
0343      DATA      BYTE_COUNT(1) /*SBI*/
0344      DATA      BYTE_COUNT(2) /*DDI*/
0345
0346
0347      CHARACTER*15      SILO_DIR
0348      DATA      SILO_DIR /*SILO DIRECTION */
0349
0350
0351      CHARACTER*1      V1DR_WD(1:6)
0352      DATA      V1DR_WD(1) /*2*/
0353      DATA      V1DR_WD(2) /* */
0354      DATA      V1DR_WD(3) /*1*/
0355      DATA      V1DR_WD(4) /*1*/
0356      DATA      V1DR_WD(5) /*2*/
0357      DATA      V1DR_WD(6) /* */
0358
0359
0360      CHARACTER*1      V1DR_ID(1:6)
0361      DATA      V1DR_ID(1) /*1*/
0362      DATA      V1DR_ID(2) /* */
0363      DATA      V1DR_ID(3) /*1*/
0364      DATA      V1DR_ID(4) /*2*/
0365      DATA      V1DR_ID(5) /*2*/
0366      DATA      V1DR_ID(6) /* */
0367
0368
0369
0370      CALL FRCTOF (LUN)
0371
0372      call dhead1 (lun,'DR32')
0373
0374      C
0375      C      DR780 CONTROL REGISTER
0376      C
0377
0378      CALL DR780_REGA (LUN,DR_CR)
0379
```

```
0380 C
0381 C      DR780 UTILITY REGISTER
0382 C
0383
0384 CALL LINCHK (LUN,1)
0385
0386 WRITE(LUN,25) DR_UTL
0387 25  FORMAT(' ',T8,'DR_UTL',T24,Z8.8)
0388
0389 FIELD = LIB$EXTZV(0,8,DR_UTL)
0390
0391 IF (FIELD .LE. 'FB'X) THEN
0392
0393 CALL LINCHK (LUN,2)
0394
0395 C
0396 C      2's complement value and get data rate in Mbytes.
0397 C      The value 5 (maximum) = 8 Mbytes.
0398 C
0399
0400 field = jidim(256,field)
0401
0402 C
0403 C      Use Table 3-9 dr780 users guide formulae to calculate data rate
0404 C
0405
0406 float = floatj(field)
0407
0408 float = 40/float
0409
0410 C
0411 C      Calculate clock period. Since 32 bits are transfered at a time
0412 C      then the number of micro-seconds in one second are divided by
0413 C      the number of bytes transfered in one second to obtain the clock
0414 C      period.
0415 C
0416
0417 float = (1024*1024) / ((float*1024*1024)/4)
0418
0419 WRITE(LUN,26) FLOAT
0420 26  FORMAT(' ',T40,'PERIOD = ',F<COMPRESSF (FLOAT,1)>.1,'. MICRO/SEC')
0421
0422 C
0423 C      Calculate the Kbyte data rate.
0424 C
0425
0426 float = floatj(field)
0427
0428 float = (40/float)*1024
0429
0430 WRITE(LUN,27) FLOAT
0431 27  FORMAT(' ',T40,'TRANSFER RATE = ',
0432          1 F<COMPRESSF (FLOAT,2)>.2,'. KB/SEC')
0433 ENDIF
0434
0435 CALL OUTPUT (LUN,DR_UTL,V1DR_UTL,11,11,11,'0')
0436
```


0437 CALL OUTPUT (LUN,DR_UTL,V2DR_UTL,25,25,31,'0')

0438
0439 C
0440 C DR780 WCS REGISTERS
0441 C

0442
0443 DO 40 I = 1,3
0444
0445 FIELD = LIB\$EXTZV(0,31,DR_WCS(I))
0446
0447 IF (FIELD .NE. 0) THEN
0448
0449 CALL LINCHK (LUN,1)
0450
0451 WRITE(LUN,30) I,DR_WCS(I)
0452 30 FORMAT(' ',T8,'DR WCS',I1,T24,Z8.8)
0453 ENDIF

0454
0455 40 CONTINUE
0456

0457 C
0458 C DR780 SBI ADDRESS REGISTER
0459 C

0460
0461 CALL LINCHK (LUN,1)
0462
0463 WRITE(LUN,50) DR_BA
0464 50 FORMAT(' ',T8,'DR BA',T24,Z8.8)

0465
0466 FIELD = LIB\$EXTZV(28,2,DR_BA)
0467
0468 CALL LINCHK (LUN,1)

0469
0470 IF (FIELD .EQ. 0) THEN
0471
0472 WRITE(LUN,65) SILO_DIR
0473 65 FORMAT(' ',T40,A15,'= SBI READ')

0474
0475 ELSE IF (FIELD .EQ. 3) THEN

0476
0477 WRITE(LUN,70) SILO_DIR
0478 70 FORMAT(' ',T40,A15,'= SBI WRITE')
0479 ENDIF

0480
0481 C
0482 C DR780 BYTE COUNT REGISTERS
0483 C

0484
0485 DO 100 I = 1,2
0486

0487 CALL LINCHK (LUN,2)
0488
0489 WRITE(LUN,80) I,DR_BC(I),BYTE_COUNT(I)
0490 80 FORMAT(' ',T8,'DR BC',I1,T24,Z8.8,/,
0491 1 T40,A3,' BYTE COUNT')

0492
0493 100 CONTINUE

```
0494
0495 C
0496 C
0497 C
0498
0499
0500
0501 DR780 COMMAND ADDRESS REGISTERS
0502
0503 DO 150 I = 1,2
0504
0505 FIELD = LIB$EXTZV(0,28,DR_CA(I))
0506
0507 FIELD = JISHFT(FIELD,2)
0508
0509 CALL LINCHK (LUN,3)
0510
0511
0512 WRITE(LUN,120) I,DR_CA(I),I,FIELD
0513 FORMAT(' ',T8,'DR CA',I1,T24,Z8.8,/,
0514 1 T40,'LAST COMMAND ADDRESS EXECUTED ID',I1,/,
0515 2 T40,'LONGWORD ADDRESS = ',Z8.8)
0516
0517 FIELD = LIB$EXTZV(28,4,DR_CA(I))
0518
0519 CALL SBI_COMMAND (LUN,FIELD)
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539 150 CONTINUE
0540
0541 C
0542 C
0543 C
0544 DR780 WRITE DATA REGISTERS
0545
0546
0547 DO 165,J = 1,2
0548
0549 FIELD = LIB$EXTZV(28,4,DR_CA(J))
0550
0551 IF (FIELD .EQ. SBI_FUNC2
0552 1 .OR.
0553 2 FIELD .EQ. SBI_FUNC7
0554 3 .OR.
0555 4 FIELD .EQ. SBI_FUNC11) THEN
0556
0557 DO 160,I = J*2,(J*2) + 2
0558
0559 IF (V1DR_WD(I) .NE. '*') THEN
0560
0561 CALL LINCHK (LUN,2)
0562
0563 WRITE(LUN,155) I,DR_WD(I),V1DR_WD(I),V1DR_ID(I)
0564 FORMAT(' ',T8,'DR WD',I1,T24,Z8.8,/,
0565 1 T40,'WRITE DATA',A1,' ID',A1)
0566
0567 ENDF
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000
```



```
0551
0552      CALL LINCHK (LUN,1)
0553
0554      WRITE(LUN,180) DR_SL
0555 180    FORMAT(' ',T8,'DR-SL',T24,Z8.8)
0556
0557      CALL OUTPUT (LUN,DR_SL,V1DR_SL,0,0,14,'0')
0558
0559      FIELD = LIB$EXTZV(16,8,DR_SL)
0560
0561      IF (FIELD .NE. 0) THEN
0562
0563      CALL LINCHK (LUN,3)
0564
0565      WRITE(LUN,182) ('*',I = 0,32)
0566 182    FORMAT(' ',T40,33A1,/,T47,'FAR-END STATUS BYTE',/)
0567
0568      DO 186,I = 0,4
0569
0570      FIELD1 = LIB$EXTZV(I,1,FIELD)
0571
0572      IF (FIELD1 .NE. 0) THEN
0573
0574      CALL LINCHK (LUN,1)
0575
0576      WRITE(LUN,184) I
0577 184    FORMAT(' ',T40,'FAR-END DEVICE STATUS BIT ',I1,'. SET')
0578      ENDIF
0579
0580 186    CONTINUE
0581
0582      CALL OUTPUT (LUN,DR_SL,V2DR_SL,21,21,23,'0')
0583
0584      CALL LINCHK (LUN,1)
0585
0586      WRITE(LUN,188) ('*',I = 0,32)
0587 188    FORMAT(' ',T40,33A1)
0588      ENDIF
0589
0590  C
0591  C      DR780 SYSTEM BASE REGISTER
0592  C
0593
0594      CALL LINCHK (LUN,2)
0595
0596      WRITE(LUN,190) DR_SBR
0597 190    FORMAT(' ',T8,'DR-SBR',T24,Z8.8,/,
0598      1 T40,'DR SYSTEM BASE REG (PHY ADDR)')
0599
0600  C
0601  C      DR780 GLOBAL BASE REGISTER
0602  C
0603
0604      CALL LINCHK (LUN,2)
0605
0606      WRITE(LUN,200) DR_GBR
0607 200    FORMAT(' ',T8,'DR-GBR',T24,Z8.8,/,
```

DR780

M 10
16-Sep-1984 00:01:25
5-Sep-1984 13:53:26VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]DR780.FOR;1

Page 8

```
0608      1 T40,'DR GLOBAL BASE REG (VIRT ADDR)')
0609
0610      C
0611      C      DR780 COMMAND BLOCK INFORMATION
0612      C
0613
0614      CALL LINCHK (LUN,3)
0615
0616      WRITE(LUN,215)
0617 215    FORMAT(/' ', 'DR CURRENT COMMAND BLOCK',/)
0618
0619      DO 220 I = 1,3
0620
0621      CALL LINCHK (LUN,2)
0622
0623      WRITE(LUN,210) I,DR_CB(I),V1DR_CB(I)
0624 210    FORMAT(' ',T8,'DR CB',I1,T24,Z8.8,/,
0625             1 T40,A<COMPRESSC (V1DR_CB(I))>)
0626
0627 220    CONTINUE
0628
0629      C
0630      C      DR780 BUFFER BLOCK INFORMATION
0631      C
0632
0633      CALL LINCHK (LUN,3)
0634
0635      WRITE(LUN,225)
0636 225    FORMAT(/' ', 'DR CURRENT BUFFER BLOCK',/)
0637
0638      DO 250,I = 1,3
0639
0640      CALL LINCHK (LUN,2)
0641
0642      WRITE(LUN,240) I,DR_BB(I),V1DR_BB(I)
0643 240    FORMAT(' ',T8,'DR BB',I1,T24,Z8.8,/,
0644             1 T40,A<COMPRESSC (V1DR_BB(I))>)
0645
0646 250    CONTINUE
0647
0648      C
0649      C      DR780 'GENERAL PURPOSE REGISTERS'
0650      C
0651
0652      CALL LINCHK (LUN,3)
0653
0654      WRITE(LUN,265)
0655 265    FORMAT(/' ', 'DR GENERAL PURPOSE REGISTERS',/)
0656
0657      DO 280,I = 0,8
0658
0659      CALL LINCHK (LUN,1)
0660
0661      WRITE(LUN,270) I,DR_GPR(I)
0662 270    FORMAT(' ',T8,'DR R',I1,T24,Z8.8)
0663
0664 280    CONTINUE
```



```
0665
0666      CALL LINCHK (LUN,2)
0667
0668      WRITE(LUN,290) DR_GPR(9)
0669 290   FORMAT(' ',T8,'DR-R9',T24,Z8.8,/,
0670          1 T40,'PAGE COUNT')
0671
0672      CALL LINCHK (LUN,2)
0673
0674      WRITE(LUN,300) DR_GPR(10)
0675 300   FORMAT(' ',T8,'DR-R11',T24,Z8.8,/,
0676          1 T40,'DDI CONTROL FLAG REGISTER')
0677
0678      CALL LINCHK (LUN,2)
0679
0680      WRITE(LUN,310) DR_GPR(11)
0681 310   FORMAT(' ',T8,'DR-R12',T24,Z8.8,/,
0682          1 T40,'COPY OF STATE REGISTER')
0683
0684      CALL LINCHK (LUN,2)
0685
0686      WRITE(LUN,320) DR_GPR(12)
0687 320   FORMAT(' ',T8,'DR-R13',T24,Z8.8,/,
0688          1 T40,'COPY OF FLAG REGISTER 0')
0689
0690      CALL LINCHK (LUN,2)
0691
0692      WRITE(LUN,330) DR_GPR(13)
0693 330   FORMAT(' ',T8,'DR-R14',T24,Z8.8,/,
0694          1 T40,'COPY OF LS.SAV')
0695
0696      CALL LINCHK (LUN,2)
0697
0698      WRITE(LUN,340) DR_GPR(14)
0699 340   FORMAT(' ',T8,'DR-R15',T24,Z8.8,/,
0700          1 T40,'MICRO STATUS REGISTER')
0701
0702      call linchk (lun,1)
0703
0704      write(lun,345)
0705 345   format(' ',:)
0706
0707      call orb$l_owner (lun,emb$l_dv_ownuic)
0708
0709      call ucb$l_char (lun,emb$l_dv_char)
0710
0711      call ucb$w_sts (lun,emb$w_dv_sts)
0712
0713      call ucb$l_opcnt (lun,emb$l_dv_opcnt)
0714
0715      call ucb$w_errcnt (lun,emb$w_dv_errcnt)
0716
0717      if (emb$w_hd_entry .ne. 98) then
0718
0719      call linchk (lun,1)
0720
0721      write(lun,345)
```

```

0722
0723      call xfdriver_qio (lun,emb$w_dv_func)
0724
0725      call irp$w_bcmt (lun,emb$w_dv_bcmt)
0726
0727      call irp$w_boff (lun,emb$w_dv_boff)
0728
0729      call irp$l_pid (lun,emb$l_dv_rapid)
0730
0731      call irp$q_iosb (lun,emb$l_dv_iosb1)
0732      endif
0733
0734      RETURN
0735      END

```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	2486	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	1061	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	956	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 DR32	600	PIC OVR REL GBL SHR NOEXE RD WRT LONG
4 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	5615	

ENTRY POINTS

Address	Type	Name
0-00000000		DR780

VARIABLES

Address	Type	Name	Address	Type	Name
2-00000110	I*4	COMPRESS4	4-00000066	I*4	DR_BA
4-00000052	I*4	DR_CR	4-0000009A	I*4	DR_GBR
4-00000096	I*4	DR_SBR	4-00000092	I*4	DR_SL
4-00000056	I*4	DR_UTL	4-0000001C	L*1	EMB\$B_DV_CLASS
4-00000010	L*1	EMB\$B_DV_ERTCNT	4-00000011	L*1	EMB\$B_DV_ERTMAX
4-0000003E	L*1	EMB\$B_DV_NAMLNG	4-0000003A	L*1	EMB\$B_DV_SLAVE
4-0000001D	L*1	EMB\$B_DV_TYPE	4-00000036	I*4	EMB\$B_DV_CHAR
4-00000012	I*4	EMB\$B_DV_IOSB1	4-00000016	I*4	EMB\$B_DV_IOSB2
4-00000026	I*4	EMB\$B_DV_MEDIA	4-0000004E	I*4	EMB\$B_DV_NUMREG
4-0000002E	I*4	EMB\$B_DV_OPCNT	4-00000032	I*4	EMB\$B_DV_OWNUIC
4-0000001E	I*4	EMB\$B_DV_RQPID	4-00000000	I*4	EMB\$B_DV_SID
4-0000003F	CHAR	EMB\$B_DV_NAME	4-00000024	I*2	EMB\$W_DV_BCNT
4-00000022	I*2	EMB\$W_DV_BOFF	4-0000002C	I*2	EMB\$W_DV_ERRCNT
4-0000003C	I*2	EMB\$W_DV_FUNC	4-0000001A	I*2	EMB\$W_DV_STS
4-0000002A	I*2	EMB\$W_DV_UNIT	4-00000004	I*2	EMB\$W_DV_ENTRY


```

4-0000000E I*2 EMBSW HD-ERRSEQ
2-00000108 I*4 FIELDT
2-00000114 I*4 I
AP-00000004a L*1 LUN

```

```
2-00000104 I*4 FIELD
2-0000010C R*4 FLOAT
2-00000118 I*4 J
2-000000F5 CHAR SILO_DIR
```

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-000000E3	CHAR	BYTE_COUNT	6	(2)
4-000000AA	I*4	DR_BB	12	(3)
4-0000006A	I*4	DR_BC	8	(2)
4-00000076	I*4	DR_CA	8	(2)
4-0000009E	I*4	DR_CB	12	(3)
4-000000B6	I*4	DR_GPR	64	(0:15)
4-0000005A	I*4	DR_WCS	12	(3)
4-0000007E	I*4	DR_WD	20	(5)
4-00000000	L*1	EMB	512	(0:511)
4-00000052	I*4	EMB\$-DV_REGSAB	420	(0:104)
4-00000006	I*4	EMB\$Q-HD-TIME	8	(2)
3-0000019B	CHAR	V1DR_BB	93	(3)
3-000001F8	CHAR	V1DR_CB	96	(3)
2-000000EF	CHAR	V1DR_ID	6	(6)
3-00000000	CHAR	V1DR_SL	345	(0:14)
2-00000000	CHAR	V1DR_UTL	10	(11:11)
2-000000E9	CHAR	V1DR_WD	6	(6)
3-00000159	CHAR	V2DR_SL	66	(21:23)
2-0000000A	CHAR	V2DR_UTL	217	(25:31)

LABELS

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-00000040	25'	1-00000053	26'	1-00000078	27'	1-000000A1	30'	**	40	1-000000B6	50'
1-000000C8	65'	1-000000DC	70'	1-000000F1	80'	**	100	1-00000117	120'	**	150
1-0000016D	155'	**	160	**	165	1-00000199	180'	1-000001AB	182'	1-000001CE	184'
**	186	1-000001F9	188'	1-00000203	190'	1-00000238	200'	1-0000028E	210'	1-0000026E	215'
**	220	1-000002AB	225'	1-000002CA	240'	**	250	1-000002E7	265'	1-0000030B	270'
**	280	1-0000031E	290'	1-0000033F	300'	1-00000370	310'	1-0000039E	320'	1-000003CD	330'
1-000003F3	340'	1-00000420	345'								

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name	Type	Name	Type	Name		
I*4	COMPRESSC IRPSQ_IOSB OUTPUT XFDRIVER_QIO	I*4	COMPRESSF IRPSW_BCNT SBI_COMMAND		DHEAD1 IRPSW_BOFF UCBSL_CHAR	I*4	DR780_REGA LIB\$EXTZV UCBSL_OPCNT		FRCTOF LINCHK UCBSW_ERRCNT		IRPSL_PID ORBSL_OWNER UCBSW_STS

0001
0002
0003
0004
0005
0006
0007
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305

Subroutine XFDRIVER_QIO (lun,emb\$w_dv_func)

include 'src\$:qiocommon.for /nolist'

byte lun

integer*2 emb\$w_dv_func

integer*4 qiocode(0:1,0:63)

if (qiocode(0,0) .eq. 0) then

qiocode(1,01) = %loc(io\$_loadmcode)

qiocode(1,06) = %loc(io\$_startdatap)

qiocode(1,56) = %loc(io\$_startdata)

do 10,i = 0,63

qiocode(0,i) = 33

if (qiocode(1,i) .eq. 0) then

qiocode(1,i) = %loc(qio_string)

endif

10

continue

endif

call irp\$w_func (lun,emb\$w_dv_func,
1 qiocode(0,lib\$extzv(0,6,emb\$w_dv_func)))

return

end

PROGRAM SECTIONS

Name	Bytes	Attributes							
0 \$CODE	119	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT LONG
1 \$PDATA	8	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT LONG
2 \$LOCAL	548	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT LONG
3 Q1OCOMMON	1247	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT LONG
Total Space Allocated		1922							

ENTRY POINTS

Address	Type	Name
0-00000000		XFDRIVER_Q10

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008a	I*2	EMBSW DV_FUNC	2-00000200	I*4	I
3-00000442	CHAR	IOS_ABORT	3-00000340	CHAR	IOS_ACCESS
3-000003C2	CHAR	IOS_ACPCONTROL	3-000004B3	CHAR	IOS_AVAILABLE
3-00000297	CHAR	IOS_CLEAN	3-00000369	CHAR	IOS_CREATE
3-00000385	CHAR	IOS_DEACCESS	3-00000393	CHAR	IOS_DELETE
3-0000026D	CHAR	IOS_DIAGNOSE	3-00000065	CHAR	IOS_DRVCLR
3-000004CB	CHAR	IOS_DSE	3-000000A9	CHAR	IOS_ERASETAPE
3-00000276	CHAR	IOS_FORMAT	3-00000071	CHAR	IOS_INITIALIZE
3-00000014	CHAR	IOS_LOADMCODE	3-000003A1	CHAR	IOS_MODIFY
3-000003E2	CHAR	IOS_MOUNT	3-00000000	CHAR	IOS_NOP
3-0000009D	CHAR	IOS_OFFSET	3-000000EB	CHAR	IOS_PACKACK
3-000000E0	CHAR	IOS_QSTOP	3-000003EF	CHAR	IOS_RDSTATS
3-00000421	CHAR	IOS_READCSR	3-00000169	CHAR	IOS_READHEAD
3-000002B6	CHAR	IOS_READLBLK	3-0000013F	CHAR	IOS_READPBLK
3-00000200	CHAR	IOS_READPRESET	3-00000195	CHAR	IOS_READTRACKD
3-0000033A	CHAR	IOS_READVBLK	3-0000045A	CHAR	IOS_READWTHBUF
3-00000484	CHAR	IOS_READWTHXBUF	3-0000004D	CHAR	IOS_RECAL
3-0000007C	CHAR	IOS_RELEASE	3-000001AB	CHAR	IOS_REREADN
3-000001B8	CHAR	IOS_REREADP	3-000000CA	CHAR	IOS_RETCENTER
3-000002E6	CHAR	IOS_REWIND	3-000002C9	CHAR	IOS_REWINDOFF
3-000000FC	CHAR	IOS_SEARCH	3-00000024	CHAR	IOS_SEEK
3-00000231	CHAR	IOS_SENSECHAR	3-00000309	CHAR	IOS_SENSEMODE
3-0000021D	CHAR	IOS_SETCHAR	3-000003B8	CHAR	IOS_SETCLOCK
3-00000088	CHAR	IOS_SETCLOCKP	3-000002DD	CHAR	IOS_SETMODE
3-000002ED	CHAR	IOS_SKIPFILE	3-000002FA	CHAR	IOS_SKIPRECORD
3-00000029	CHAR	IOS_SPACEFILE	3-0000010E	CHAR	IOS_SPACERECORD
3-000003D7	CHAR	IOS_STARTDATA	3-000000B4	CHAR	IOS_STARTDATAP
3-00000037	CHAR	IOS_STARTMPROC	3-0000020F	CHAR	IOS_STARTSPNDL
3-00000059	CHAR	IOS_STOP	3-0000000D	CHAR	IOS_UNLOAD
3-0000046B	CHAR	IOS_WRITEBUFNCRC	3-0000011E	CHAR	IOS_WRITECHECK
3-000001E4	CHAR	IOS_WRITECHECKH	3-000003FF	CHAR	IOS_WRITECSR
3-00000153	CHAR	IOS_WRITEHEAD	3-000002A2	CHAR	IOS_WRITELBLK
3-00000247	CHAR	IOS_WRITEMARK	3-00000314	CHAR	IOS_WRITEOF
3-0000012A	CHAR	IOS_WRITEPBLK	3-000001C9	CHAR	IOS_WRITERET

XFDRIVER_QIO

F 11
16-Sep-1984 00:01:25
5-Sep-1984 13:53:26

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]DR780.FOR;1

Page 14

3-0000017E CHAR IOS_WRIETACKD
3-00000448 CHAR IOS_WRIETWTHBUF
AP-00000004a L*1 LUN

3-00000326 CHAR IOS_WRITEVBLK
3-00000257 CHAR IOS_WRTTMKR
3-000004A1 CHAR QIO_STRING

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000000	I*4	QIOCODE	512	(0:1, 0:63)

LABELS

Address	Label
**	10

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
	IRPSW_FUNC	I*4	LIB\$EXTZV

COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:DR780/OBJ=OBJ\$:DR780 MSRC\$:DR780

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)

/DEBUG=(NOSYMBOLS,TRACEBACK)

/STANDARD=(NOSYNTAX,NOSOURCE FORM)

/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)

/F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time:	8.91 seconds
Elapsed Time:	19.95 seconds
Page Faults:	264
Dynamic Memory:	236 pages

0147

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY